AD-A047 530 AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA) F/G 9/2
A GENERAL PURPOSE OUTPUT PROGRAM FOR USE IN SIMULATION,(U)
DEC 76 P G NANKIVELL, N E GILBERT
UNCLASSIFIED ARL/AERO NOTE-367 NL

1 OF 1
AD
A047530

END
DATE
FILMED
1-78
DDC

ARL/Aero. Note 367

AR No 236

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

Aerodynamics Note 367

A GENERAL PURPOSE OUTPUT PROGRAM FOR USE IN SIMULATION

by

P. G. NANKIVELL and N. E. GILBERT

DECEMBER 1976

AR No. 236

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

AERODYNAMICS NOTE 367

A GENERAL PURPOSE OUTPUT PROGRAM FOR USE IN SIMULATION

by

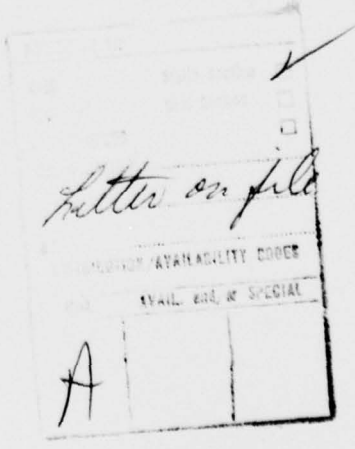
P. G. NANKIVELL and N. E. GILBERT

SUMMARY

The 'block oriented' simulation language CSMP-10(ARL) consists of a modelling program, which performs the model simulation and stores the output in a binary file, and an output program, which prints and plots the character conversion of the binary file. The output program which is described here, is written mainly in FORTRAN IV for a PDP-10 computer and is controlled interactively from a Teletype. Although the output program has been written primarily to be used in conjunction with the modelling program, it may also be used as a general purpose output program by using a supplied subprogram package. The program provides Teletype and line printer output in either tabular or graphical form and incremental plotter output in the form of 'strip' or 'overlay' plots.

POSTAL ADDRESS: Chief Superintendent, Aeronautical Research Laboratories,
Box 4331, P.O., Melbourne, Victoria, 3001, Australia.

CONTENTS



1. INTRODUCTION

When performing computer simulations of continuous and discrete dynamic systems, large amounts of output data are often generated and presented as tables or plots of variables as functions of time. Since much of the program writing is concerned with outputting the data, simulation languages, with their generalised output facilities, offer many advantages to the programmer, both in convenience and economies of effort. However, an ideal form of output is not always possible because of limitations in the variety of output presentation.

In developing a modified version of the 'block oriented' simulation language CSMP-10,[1] called CSMP-10(ARL),[2] one of the major changes made has been to divide the language into a modelling program named BOMMP (Block Oriented Mathematical Modelling Program) and an output program named TRANS (Translation). The modelling program is used to perform the simulation and store the output, at the completion of each time step, in binary form on a specified storage device. The output program is used to print and plot the character conversion of the binary file in either tabular or graphical form. In CSMP-10, which was developed from CSMP-9,[3] a significant amount of core storage is used to code the output section of the program. By dividing the language into two parts, the amount of core storage available for each process is significantly increased. This allowed more facilities to be included in both the modelling and output section of the language as well as increasing the size of the problem that can be handled. Because the 'raw' output data are stored in a file in binary form by the modelling program, various forms of output of the same data may be obtained without repeating the simulation. A saving of storage is made in the binary file by storing an output value only if it changes by more than a prescribed percentage from the value last stored.

In Reference 2, which fully describes the modelling program, only a brief account of running the output program is given; a full description is given here. Although the output program has been written primarily to be used in conjunction with the modelling program, the output program may also be used independently as a general purpose output program, provided data are supplied in the appropriate format. As with BOMMP, the program TRANS is designed to be run interactively from a Teletype.

The output program is written mainly in FORTRAN IV for the PDP-10 FORTRAN compiler 'F40 Version-27' running under the operating system 'FOROTS'. The program provides Teletype and line printer output in either tabular or graphical form and incremental plotter output in the form of 'strip' plots (of the form produced by a multi-channel chart recorder) and 'overlay' plots (a single graph for up to eight output variables).

The output file obtained by running BOMMP, which becomes the input file for TRANS, includes the time limits and interval, a title, and labels identifying all the variables. Subject to some restrictions, these may be redefined. For graphical output, convenient scale limits are calculated by TRANS; however, these limits may be defined by the user.

The basic operations of the output program necessary to produce Teletype output are given in Section 2, and in Section 3 are described the operations which enable the user to

(*a*) obtain line printer and incremental plotter output,

(*b*) redefine the time parameters and labelling information, and

(*c*) define the scale limits used for graphical output.

With the aid of a complete example, Section 4 shows how the output program may be used without running the modelling program, and in Section 5 a description is given of the structure of the computer program and the input file.

2. BASIC OPERATIONS OF OUTPUT PROGRAM

This section explains how, using basic operations, the output program may be run to produce Teletype output. Much of this section is contained in Reference 2, but is included here

for completeness of presentation. In the block oriented language CSMP-10(ARL), the output variables are referred to as blocks, and each block is identified by a number. The file containing the output values of these blocks is therefore referred to as the block output file to distinguish it from the model output file, which is used by the modelling program to store the various statements specifying the current model.[2] The output (stored in the block output file named OUTPT.DAT) produced by the non-linear spring problem of Reference 2 is used below to illustrate the operations of TRANS. Errors and their diagnostic messages are presented in Appendix A.

2.1 Setting up the Program

For the PDP-10 computer the relocatable binary file TRANS.REL is first loaded and a core image (TRANS.SAV) is saved. Execution of the program is then initiated by the user and the filename (without extension) of the block output file produced by running BOMMP is typed by the user. The filename, which is OUTPT for the example shown, has a maximum of five characters. The output program then types the title, date and time of creation of the block output file, integration interval, run CPU time, and time parameters, followed by an '*', which signifies the command mode (see below). An example of how to set up the program in the above manner is shown below:†

.RU TRANS

I/P FILENAME = **OUTPT** Extension name 'DAT' is assumed

NON-LINEAR SPRING WITH DASHPOT

I/P FILE RECORDED ON 21-APR-76 AT 16:57

INTEGN INT = 1.0000E−01; RUN CPU TIME = 1.25 SEC.

TIME FROM 0.0000E−01 TO 2.0000E+01 IN STEPS OF 1.0000E+00

*

Although the filename extension is not specified by the user, the extension name 'DAT' is assumed. If, in the above example, a file named OUTPT.DAT is not found on the disk, the message

OUTPT.DAT NOT ON DISK

I/P FILENAME =

is typed by the program so that a new filename may be specified.

As with the modelling program, the output program is controlled by commands, which are typed by the user whenever an '*' is typed by the program (i.e. whenever control is at the command mode). The commands may be shortened to three characters and are listed with brief descriptions in Table 1.

By using the appropriate commands, one or more particular types of output may now be specified. To process the input file, producing tabular or graphical output, the 'GOE' command should follow. Control is then returned to the command mode. Repetition of the appropriate command for a particular type of output before the 'GOE' command, deletes the effect of the

† Teletype messages typed by the user are shown in bold upper case; messages type by the computer program are shown in italic upper case. Comments on Teletype messages are shown in upper and lower case print alongside the appropriate message.

TABLE 1

Output Program Commands

All commands may be shortened to three characters

Command	Effect	Section(s) for Reference
EXIT	Returns control to the monitor	2.1
GOE	Processes the input file to obtain tabular or graphical output for the commands 'PRCOLUMN', 'PRPLOT', 'PLSTRIP', and 'PLOVERLAY'	2.1
LABEL	Enables modification and addition to labelling information	3.2
PLOVERLAY	Specifies 'overlay' plotting on the incremental plotter	3.5
PLSTRIP	Specifies 'strip' plotting on the incremental plotter	3.5
PRCOLUMN	Specifies tabular output on the line printer or Teletype in the form of printed columns	2.2, 3.4
PRPLOT	Specifies graphical output on the line printer or Teletype	2.2, 3.4
RUN	Equivalent to the commands (in order) 'PRCOLUMN', 'PRPLOT', 'PLSTRIP', and 'GOE'; if any form of output is not required, a 'carriage-return' is typed in place of the block numbers	3.6
SCALE	Enables the user to specify scale limits used for graphical output	3.3
TIME	Redefines time parameters	3.1

previous use of that command; repetition after the 'GOE' command, results in additional output. Control is returned to the monitor by the 'EXIT' command.

2.2 Teletype Output

The commands 'PRCOLUMN' and 'PRPLOT' are used to obtain tabular and graphical output respectively on either the line printer or Teletype. On the graphical output, line spaces correspond to equal increments in time (or dependent variable), and column spaces correspond to increments in the output variables. For each of these commands, the required block numbers, which are separated by any non-numeric character, are first listed. They may be typed on more than one line, each line consisting of a maximum of 72 characters, and the list is terminated by two consecutive 'carriage-returns'. If all the block numbers are required, the user may type **A** in place of the individual block numbers, in which case a single 'carriage-return' terminates the list. When specifying blocks by their individual block numbers, a maximum of a hundred blocks may be processed each time the 'GOE' command is used; however, all blocks are processed when the user types **A** instead, even if there are more than a hundred blocks. Following the Teletype message

IS O/P TO TTY REQRD :

the user types **Y** if the output is to be printed directly on the Teletype, or any other character, including a 'carriage-return', if the output is to be stored on a disk file so that the resulting file can subsequently be printed on the line printer (see Section 3.4). This latter form of output

contains more information than the Teletype output. On completing the Teletype responses to either the 'PRCOLUMN' or 'PRPLOT' commands, the program returns control to the command mode. The 'GOE' command is then used to obtain the Teletype output. It should be noted that both tabular and graphical output cannot be obtained on the Teletype by using a single 'GOE' command following the 'PRCOLUMN' and 'PRPLOT' commands.

All values printed on both forms of Teletype output are shown to three significant figures. For the Teletype tabular output, up to six variables, in addition to the time, are printed in columns on the same table. If more than six are to be printed, six at a time (or the remainder if the last group is less than six) are printed in separate tables. For the Teletype graphical output, up to five variables, represented by the symbols '+', '*', '@', '×', and '#', are displayed on the same graph with a width of 50 character spaces per line (i.e. 51 distinct points). Where two or more symbols coincide, the one appearing last in the above list is shown. If more than five variables are to be plotted, five at a time (or the remainder if the last group is less than five) are plotted on separate graphs. Convenient plotting scale limits are calculated by the output program and are shown at the head of each Teletype plot (see Section 3.3 for specification of limits by the user). The output is printed or plotted over the full time interval for which data are stored in the output file, unless the 'TIME' command is used to restrict the time interval (see Section 3.1).

The following example shows how both tabular and graphical Teletype output may be obtained for the non-linear spring problem:

***PRC**

PRINTING IN COLUMNS :

BLKS
A List terminated by a single 'carriage-return'

IS O/P TO TTY REQRD : **Y**

***GOE**

*** RUNNING ***

(See Section 1 of Appendix B for Teletype output)

***PRP**

PRINTER PLOTS :

BLKS
9, 4 List terminated by two consecutive 'carriage-returns'

IS O/P TO TTY REQRD : **Y**

***GOE**

*** RUNNING ***

(See Section 2 of Appendix B for Teletype output)

***EXI**

END OF EXECUTION
CPU TIME 2.27: ELAPSED TIME: 9:9.82
EXIT

3. FURTHER OPERATIONS OF OUTPUT PROGRAM

The basic operations necessary to produce Teletype output only were described in the previous section. This section describes operations enabling the user to

(*a*) obtain line printer and incremental plotter output,
(*b*) redefine the time parameters and labelling information, and
(*c*) define the scales used for graphical output.

The non-linear spring problem output is again used as an example, but the output file OUTPT.DAT is obtained by running BOMMP with an output interval of 0.1 (previously 1.0); this allows smoother curves for output on the incremental plotter. Various files are created by the output program to store labelling, scaling, or output information. The labelling and scaling information may be read in when re-running the program, and the output information enables printed and plotted output to be obtained appropriately on the line printer or incremental plotter. The labelling and scaling files are named TRANS.LBL and TRANS.SCL respectively. The files containing output information are of the form "filename"."extension", where "filename" is the filename (without extension) of the block output file produced by running BOMMP (i.e. OUTPT for the present example), and "extension" is set by TRANS for each form of output. Table 2 provides a summary of the disk files created by TRANS on using the appropriate commands.

TABLE 2

Disk Files Associated with Commands

Files containing output information for subsequent printing or plotting have the filename (without extension) of the block output file produced by running BOMMP (i.e. OUTPT for the present example)

Command	Name of File	Logical Unit Number
LABEL	TRANS.LBL	2
PLOVERLAY, PLSTRIP	OUTPT.PLT	9
PRCOLUMN	OUTPT.COL	11
PRPLOT	OUTPT.PPL	10
SCALE	TRANS.SCL	7

3.1 Time Parameters

Preceding the output values stored in the block output file produced by running BOMMP are the lower and upper time values and the output time interval (see Section 5.1). The number of time values for which printed or plotted output is required may be reduced by redefining these parameters using the 'TIME' command. For the incremental plotter though, the 'TIME' command has no effect on the interval used for plotting. When using the 'TIME' command, times are always expressed in the units used in the block output file. For the present example, the lower and upper time values of 0 and 20 respectively are unchanged, and the output interval of 0.1 is altered to 0.5 as follows:

***TIM**

TIME PARAMS: LOWER, UPPER, INTERVAL = **0, 20, 0.5**

*

3.2 Labelling

Various parameters and labelling information are stored at the beginning of the block output file produced by running BOMMP (see Section 5.1). On reading in this file, TRANS stores the labelling information so that the various forms of line printer and incremental plotter output produced by TRANS may be suitably labelled. The labelling information includes the modelling program title of one line of sixty characters and a list of block labels consisting of one line of ten characters for each block. The first of these block labels is the independent variable label (referred to here as X label) of 'TIME'. A block number of −1 is used to identify this particular label.† Subsequent block labels (referred to here as Y labels) are used to identify the block variables (i.e. dependent variables). If no label is specified when running BOMMP, then by default, TRANS provides a label of the form 'BLK # n', where n is the appropriate block number. For graphical output, blocks are identified by their label, rather than by their number as is the case with Teletype output. Both labels and block numbers are shown for line printer tabular output.

The 'LABEL' command enables the title and labels to be redefined and allows the specification of an additional line for each. The title and labels, whether modified or not, are stored in a file named TRANS.LBL for future use. On using the 'LABEL' command, the labelling information required for a particular graphical or tabular output may then be read in from this file. Labels may also be defined for blocks whose output values are not read in from the block output file; this enables the user to set up a general labelling file for use with a number of different block output files which each have a number of variables in common.

Each time the 'LABEL' command is used, the output program processes three separate stages. In order of execution, these are termed the (*a*) 'source', (*b*) '*listing*', and (*c*) '*modification*' stages. The start of each of these stages corresponds to the following Teletype messages typed by TRANS

(*a*) *IS LABELLING TO BE READ FROM DSK :*

(*b*) *IS TTY LISTING REQRD :*

(*c*) *ARE MODIFICATIONS REQRD :*

The user answers each message appropriately by typing **Y** for 'yes' or any other character for 'no'.

For the 'source' stage, if the user responds by typing **Y**, the labelling information is read in from a file named TRANS.LBL; otherwise, the labelling information already stored by TRANS is used.

If 'listing' is required, the program first types the title followed by the message

BLKS FOR LISTING

The user then specifies the appropriate block numbers, either individually on a single line consisting of a maximum of 72 characters or by typing **A** if all the block numbers are required. In each case, a single 'carriage-return' terminates the list. If individual block numbers are specified, following the listing of the corresponding labels, the above Teletype message is repeated so that further block numbers may be specified. This process is repeated until the user types a 'carriage-return' only in place of a list of block numbers. If all the block numbers are specified (i.e. by typing **A**), the listing section is automatically terminated following the complete listing of the labels.

If 'modifications' are required, then for each line of labelling information to be modified, the user types the block number followed by the line number and labelling information text. For the labels, the block number is set equal to the number used to identify the corresponding output value, but for the title, the block number is set equal to zero. For both the title and labels, the line number is set equal to one if the labelling information of one line read in from the block

---

† In the modelling program, block number 1 is reserved for the time or independent variable. Because the output program may be used independently of the modelling program (see Section 4), TRANS identifies block number −1 as the time or X value.

output file is to be modified, and is set equal to two if the additional line is to be specified or modified. When the modifications are completed, control is returned to the command mode by typing an additional 'carriage-return'.

The two examples given below show how (1) the title and labels stored in the block output file OUTPT.DAT are first listed on the Teletype and then modified, and (2) by repeating the 'LABEL' command during either the same run or a subsequent one, the modified title and labels are read in from the file TRANS.LBL and listed on the Teletype. In Example 2, the same listing would be obtained without reading the labelling information from TRANS.LBL (i.e. the user types **N** to the above message (*a*)) if the 'LABEL' command was repeated during the same run.

Example 1

***LAB**

BLK NO. −1 DENOTES INDEP VARIABLE

IS LABELLING TO BE READ FROM DSK : **N**

IS TTY LISTING REQRD : **Y**

TITLE
NON-LINEAR SPRING WITH DASHPOT

Additional line is blank

BLKS FOR LISTING
A

By typing **A**, 'listing' stage automatically terminated following complete listing

BLK LABEL
−1 TIME

Additional line for each label is blank

9 Y

4 Y DBLE DOT

48 Y DOT

10 BLK# 10

ARE MODIFICATIONS REQRD : **Y**

BLK (0 FOR TITLE), LINE, TEXT
0, 1, NON-LINEAR SPRING WITH DASHPOT
0, 2, DATA (STORED IN OUTPT.DAT) PRODUCED BY RUNNING BOMMP
−1, 2, SEC
4, 2, FT/SEC/SEC
9, 2, FT
48, 2, FT/SEC

Additional 'carriage-return' terminates the list

*

Example 2

***LAB**

BLK NO. –1 DENOTES INDEP VARIABLE

IS LABELLING TO BE READ FROM DSK : **Y**

IS TTY LISTING REQRD : **Y**

TITLE

NON-LINEAR SPRING WITH DASHPOT
DATA (STORED IN OUTPT.DAT) PRODUCED BY RUNNING BOMMP

BLKS FOR LISTING
–1, 4, 9, 48

Single 'carriage-return' terminates the list

BLK LABEL
–1 TIME
SEC
4 Y DBLE DOT
FT/SEC/SEC
9 Y
FT
48 Y DOT
FT/SEC

BLKS FOR LISTING

ARE MODIFICATIONS REQRD : **N**

'Carriage-return' at beginning of line terminates 'listing' stage

*

The file TRANS.LBL may be created or modified without running TRANS. A listing of TRANS.LBL created in Example 2 above is given below to illustrate the exact form of the file; the format statement is FORMAT(I4, 1X, 12A5) for each line of the title, and FORMAT(I4, 1X, 4A5) for each label:

```
 0                  NON-LINEAR SPRING WITH DASHPOT
 0     DATA (STORED IN OUTPT.DAT) PRODUCED BY RUNNING BOMMP
-1    TIME       SEC
 9     Y          FT
 4 Y DBLE DOTFT/SEC/SEC
48   Y DOT       FT/SEC
10   BLK#   10
```

3.3 Scaling for Graphical Output

On reading in the block output file produced by running BOMMP, the output program determines lower and upper data limits for the independent variable (referred to as X) and for each of the block variables. Whenever the 'TIME' command is used, the data limits are re-determined for the reduced number of data values. When graphical output is required, these limits are then used to calculate convenient plotting scale limits. The 'SCALE' command enables this process to be bypassed and the scale limits to be provided by the user and stored in a file named TRANS.SCL for future use. Data limits determined by TRANS on reading in the block output file are not stored in TRANS.SCL. On subsequently running TRANS, the limits required

for a particular graphical output may be read in from the file TRANS.SCL on using the 'SCALE' command. As with labelling information, lower and upper limits may be defined for blocks whose output values are not always read in. When the user defines the scale limits, they are used for graphical output without any modification. Scale limits defined for the X scale are used only for incremental plots, whereas graphical output on the Teletype or line printer uses the time limits specified in the input file, or the modified limits if the 'TIME' command is used.

Use of the 'SCALE' command is very similar to that of the 'LABEL' command, and the two examples below correspond very closely to those used to illustrate the 'LABEL' command. Differences may be seen by comparing the examples. For the 'listing' stage, the limits are followed by '###' if they are data limits determined by TRANS on reading through the block output file. For modifications, on each line the user types the block number followed by the lower and upper scale limits.

Example 1

***SCA**

BLK NO. –1 DENOTES INDEP VARIABLE

ARE PLOT SCALE LIMITS TO BE READ FROM DSK : **N**

IS TTY LISTING OF LIMITS REQRD : **Y**

DENOTES DATA LIMITS ONLY (USED TO DETERMINE PLOT SCALES)

BLKS FOR LISTING
A

BLK	*LOWER*	*UPPER*	
–1	*0.0000E–01*	*2.0000E+01*	*###*
9	*–1.0000E+01*	*6.4689E+00*	*###*
4	*–8.7514E+00*	*2.0000E+01*	*###*
48	*–4.7558E+00*	*9.4473E+00*	*###*
10	*–1.0000E+02*	*4.2564E+01*	*###*

ARE MODIFICATIONS REQRD : **Y**

BLK, LOWER, UPPER
10, –60, 40
4, –10, 14

*

Example 2

***SCA**

BLK NO. –1 DENOTES INDEP VARIABLE

ARE PLOT SCALE LIMITS TO BE READ FROM DSK : **Y**

IS TTY LISTING OF LIMITS REQRD : **Y**

DENOTES DATA LIMITS ONLY (USED TO DETERMINE PLOT SCALES)

BLKS FOR LISTING
4, 10, 48

BLK	*LOWER*	*UPPER*	
4	*−1.0000E+01*	*1.4000E+01*	
10	*−6.0000E+01*	*4.0000E+01*	
48	*−4.7558E+00*	*9.4473E+00*	*# # #*

BLKS FOR LISTING

ARE MODIFICATIONS REQRD : **N**

*

As with TRANS.LBL, the file TRANS.SCL may be created or modified without running TRANS. A listing of TRANS.SCL created in Example 2 above is given below to illustrate the exact form of the file; the format statement for each line is FORMAT(I4, 2(1PE12.4)), and because the limits for only two blocks were defined, the file contains only two lines:

```
   4 −1.0000E+01  1.4000E+01
  10 −6.0000E+01  4.0000E+01
```

3.4 Line Printer Output

The commands 'PRCOLUMN' and 'PRPLOT' used to obtain tabular and graphical output respectively have been described in Section 2.2. However, only the Teletype form of output was described in detail. The line printer form of output obtained from the disk file is fully described in this section. The tabular and graphical outputs for the line printer are stored by the output program in the files "filename".COL and "filename".PPL respectively; both forms of output may therefore be obtained on using a single 'GOE' command following the 'PRCOLUMN' and 'PRPLOT' commands. On the line printer, all values printed are shown to five significant figures and labels are used to identify the blocks instead of the block numbers for graphical output, and both labels and block numbers are shown for tabular output. The title, date and time of creation of the block output file, integration interval, and run CPU time are also printed. For the tabular output, up to ten variables, in addition to the time, are printed in columns on the same table, and for the graphical output, the graph has a width of 100 character spaces per line (i.e. 101 distinct points) and symbols are overprinted by other symbols for points that coincide.

The following example shows how the files OUTPT.COL and OUTPT.PPL may be generated for the present example; the redefined time parameters, labelling information, and scales specified in Sections 3.1 to 3.3 are used:

***PRC**

PRINTING IN COLUMNS :

BLKS
A

IS O/P TO TTY REQRD : **N**

***PRP**

PRINTER PLOTS :

BLKS
A

IS O/P TO TTY REQRD : **N**

***GOE**

** *RUNNING* **

*

On subsequently using the 'EXIT' command in the above example, the files OUTPT.COL and OUTPT.PPL may be printed by running the system program 'PRINT', and are shown in Appendix C. Because carriage control characters are used extensively to produce the graphical output, the file OUTPT.PPL should be printed using the 'P' switch (on the PDP-10 computer) as follows:

.R PRINT

*17:40**OUTPT.PPL/PS
FILES PRINTED REQ. PAGES
OUTPT .PPL R01 2
EXIT

.

Carriage control characters are used minimally to produce the tabular output so that use of the 'P' switch is optional when printing the file *OUTPT.COL*.

3.5 Incremental Plotter Output

The commands 'PLSTRIP' and 'PLOVERLAY' are used to obtain 'strip' plots (of the form produced by a multi-channel chart recorder) and 'overlay' plots (a single graph for up to eight output variables). Each form of output is stored in the file "filename".PLT. Time, or the independent variable, is shown on the abscissa (X axis), and the output variables are shown on the ordinate (Y axis). As with line printer output, the title, date and time of creation of the block output file, integration interval, and run CPU time are shown on each plot. Labels are used to identify the blocks instead of the block numbers.

For both commands, the block numbers required are first listed in the same way as for the 'PRCOLUMN' and 'PRPLOT' commands (i.e. the user types **A** or specifies individual block numbers). For the 'PLOVERLAY' command only, the block numbers, whose lower and upper variable limits are used to determine the Y plotting scale are then listed in the same way. Finally, for both commands, the lengths of the X and Y axes expressed as an integer number of inches are then specified; the length of the Y axis must not exceed 8 in.

The 'strip' plots are automatically arranged in adjacent strips such that the total span of the Y axes does not exceed 8 in; further sets of plots are created until all the variables are plotted. As with graphical output on the Teletype or line printer, convenient plotting scale limits are calculated by the program for each output variable, unless the scale limits are specified by the user. For the 'overlay' plots, up to eight variables may be plotted on the same graph using a common scale. If there are more than eight, only the first eight are plotted; the remainder are ignored and an appropriate error message is typed (see Appendix A). Using only the block numbers specified for determining the Y plotting scale, the smallest lower limit and largest upper limit are determined. If both limits are scale limits specified by the user, they are used without modification. If at least one limit is a data limit, a convenient plotting scale is calculated by the program. Including the title and labels, a plot will fit within the International standard A4 paper size for an X axis of length 5 in.

The two examples given below show how for the present example, using the modifications of Sections 3.1 to 3.3, the file OUTPT.PLT may be obtained for (1) 'strip' plots and (2) 'overlay' plots. Two separate runs of TRANS were made; however, both plots may be obtained on the one file by using a 'GOE' command after each plotting command.

Example 1

***PLS**

STRIP PLOTS :

BLKS
4, 48, 9, 10 Full list of block numbers typed individually in place of **A** to obtain desired order of plots
LENGTH OF AXES IN INCHES; X, *Y* = **5, 2**

***GOE**

*** RUNNING***

*

Example 2

***PLO**

OVERLAY PLOTS :

BLKS
9, 48, 4

BLKS TO DETERMINE Y SCALE
4 List terminated by two consecutive 'carriage-returns', as in previous list
LENGTH OF AXES IN INCHES; X, Y = **5, 8**

***GOE**

*** RUNNING ***

*

On subsequently using the 'EXIT' command in each of the above examples, the file OUTPT.PLT may be plotted (see Figures 1a and 1b) by running the system program 'PLOTQ' as follows:

.R PLOTQ

*10:57****OUTPT.PLT$**
FILES PLOTTED REQ. FEET LIMITS
OUTPT .PLT R01 2 X(*8.50", 1'10.50"*), *Y*(*−5.32", 6"*)
EXIT

.

3.6 The 'RUN' Command

The 'RUN' command is equivalent to the individual commands 'PRCOLUMN', 'PRPLOT', 'PLSTRIP', and 'GOE' in the order shown. If any form of output is not required, a 'carriage-return' is typed in place of the block numbers. The use of the 'RUN' command is demonstrated by the following example, which repeats the examples of Sections 3.4 and 3.5 (except for 'overlay' plotting):

***RUN**

PRINTING IN COLUMNS :

BLKS
A

IS O/P TO TTY REQRD : **N**

PRINTER PLOTS :

BLKS
A

IS O/P TO TTY REQRD : **N**

STRIP PLOTS :

BLKS
4, 48, 9, 10

LENGTH OF AXES IN INCHES; X, Y = **5, 2**

*** RUNNING ***

*

4. USE OF OUTPUT PROGRAM WITHOUT RUNNING THE MODELLING PROGRAM

A subprogram package consisting of the FORTRAN file PAKAGE.F4 and MACRO-10 file CPU.MAC is available to enable the user to obtain an input file for TRANS without running BOMMP. An example of a main program that generates certain functions and stores the results in the file FUNCT.DAT in binary form for input to TRANS is given in Appendix D. Comments in the listing of the example main program provide a complete description on how to use this facility. The following is an example run of TRANS that uses the input file FUNCT.DAT created on running the above program:

.RU TRANS

I/P FILENAME = **FUNCT**
VARIOUS FUNCTIONS OF X

I/P FILE RECORDED ON 27-JUL-76 AT 14:02

INTEGN INT = 0.0000E−01; RUN CPU TIME = 1.35 SEC.

TIME FROM 0.0000E−01 TO 2.0000E+00 IN STEPS OF 4.0000E−03

***LAB**

BLK NO. −1 DENOTES INDEP VARIABLE

IS LABELLING TO BE READ FROM DSK : **N**

IS TTY LISTING REQRD : **Y**

TITLE

VARIOUS FUNCTIONS OF X

BLKS FOR LISTING
A

BLK	*LABEL*
−1	*X*
1	*EXP(2X)*
2	*SQUARE*
3	*SIN(8X)*
4	*X*SIN(8X)*
5	*SIN(8X**2)*
6	*SAW*
7	*NORMAL*

ARE MODIFICATIONS REQRD : **Y**

BLK (0 FOR TITLE), LINE, TEXT
2, 2, WAVE
6, 2, TOOTH
7, 2, CURVE

***SCA**

BLK NO. −1 DENOTES INDEP VARIABLE

ARE PLOT SCALE LIMITS TO BE READ FROM DSK : **N**

IS TTY LISTING OF LIMITS REQRD : **Y**

DENOTES DATA LIMITS ONLY (USED TO DETERMINE PLOT SCALES)

```
BLKS FOR LISTING
A

BLK     LOWER       UPPER
 -1   0.0000E-01  2.0000E+00 ###
  1   0.0000E-01  5.4598E+01 ###
  2  -1.0000E+00  1.0000E+00 ###
  3  -9.9738E-01  9.9777E-01 ###
  4  -1.3759E+00  1.7571E+00 ###
  5  -9.9998E-01  9.9900E-01 ###
  6   0.0000E-01  4.0000E-01 ###
  7   0.0000E-01  3.9890E-01 ###

ARE MODIFICATIONS REQRD : Y

BLK, LOWER, UPPER
1, 0, 20

*PLS

STRIP PLOTS :

BLKS
A

LENGTH OF AXES IN INCHES; X, Y = 5, 1

*GOE

** RUNNING **

*EXI


END OF EXECUTION
CPU TIME: 8.60  ELAPSED TIME: 4:49.52
EXIT

.R PLOTQ


14:24*FUNCT.PLT$
FILES PLOTTED   REQ.   FEET   LIMITS
FUNCT  .PLT     R01    2      X(8.50", 1'10.50"), Y(-5.32", 5")
EXIT

.
```

The output file FUNCT.PLT obtained above is shown in Figure 2.

In the above example, the labelling information and plotting scales were modified interactively while running TRANS. This may be inconvenient, especially if frequent use of the additional line for the title or labels is required. To avoid this inconvenience, the modifications may be effected directly in the main program by opening the files TRANS.LBL and TRANS.SCL, and storing the modified information in these files in the format specified in Sections 3.2 and 3.3. On running TRANS, the labelling information and plotting scales would then be read in using the 'LABEL' and 'SCALE' commands respectively.

5. DESCRIPTION OF OUTPUT PROGRAM

The input file structure and computer program structure are briefly described below. Further information may be obtained by referring to the FORTRAN listing.

5.1 Input File Structure

The input file for TRANS, which is generally the block output file obtained by running BOMMP, is a binary file consisting of full disk blocks of 128 words. The modelling program arranges the data into records of 126 data words with two additional words being used by the operating system 'FOROTS' to signify the beginning and end of each record.

To save storage, when running BOMMP an output value is only stored if it changes by more than a prescribed percentage from the value last stored at a previous time value. The data are therefore stored in the form of pairs consisting of a block number (stored as a real variable rather than as an integer) followed by the corresponding output value. A block number equal to -1 is used to indicate that the next word is a new time or independent variable value. For N blocks, the first $24 + 3N$ words of the input file are used to store various parameters and labelling information (see Appendix E). The next $2 + 2N$ words consist of a block number of -1, the lower time value, followed by N block number and block value pairs. Subsequently, a block number of -1, a time value, followed by a maximum of N block number and value pairs are stored, the actual number depending on whether the block values have changed sufficiently from their value last stored at a previous time step. Finally, the run CPU time is stored.

5.2 Program Structure and Storage Requirement

The computer program consists of a small main program (MAIN.) used to open the arrays that are later expanded, a number of major subroutines and various service subroutines and functions (see Appendix F). When loaded on the PDP-10 computer the program requires 13 K words of core store (1K = 1024). On running the program, a number of arrays are expanded so that an additional 8N words are required, where N is the number of blocks.

6. CONCLUDING REMARKS

Although designed to be used as the output section of the 'block oriented' simulation language CSMP-10(ARL), the output program that has been described may also be used as a general purpose output program by using a supplied program package. As with the modelling program, which is used to perform the model simulation in CSMP-10(ARL) and store the output in a binary file, the output program is written mainly in FORTRAN IV for a PDP-10 computer and is controlled interactively from a Teletype. Provision has been made for Teletype and line printer output in either tabular or graphical form and incremental plotter output in the form of 'strip' or 'overlay' plots. Facilities have been introduced that provide a large degree of flexibility in allowing the user to redefine the time parameters, labelling information, and plotting scales.

7. ACKNOWLEDGEMENT

The authors are indebted to the assistance provided by P. Gottlieb and the staff of the Computer Centre at A.R.L.

REFERENCES

1. Carnegie Mellon University — 'CSMP-10'. DECUS Program No. 10-122 (1971).

2. Gilbert, N. E. and Nankivell, P. G. — 'The Simulation Language CSMP-10(ARL)'. ARL Note Aero 362, May 1976.

3. Carnegie Mellon University — 'Block CSMP-9. A Block Oriented Continuous System Modelling Program for the PDP-9'. User Manual (1969).

POSTAL ADDRESS: Chief Superintendent, Aeronautical Research Laboratories, P.O. Box 4331, Melbourne, Victoria, 3001, Australia.

APPENDIX A

Output Program Diagnostic Error Messages

To distinguish the following error messages from the FORTRAN compiler diagnostics, all the messages below are preceded by an exclamation mark (!). An integer constant is represented by n and 's.e.' is used to abbreviate 'self-explanatory'.

! CODE PRIOR TO CALL TO EXPAND UNRECOGNIZED —obtain a new copy of the file TRANS.SAV; if error persists, keep data files and consult originator

! COMMAND ERROR —invalid program command

! END OF I/P FILE REACHED BEFORE UPPER TIME VALUE —s.e.

! FILE TRANS.LBL (CONTAINING LABELLING INFORMATION) NOT ON DSK — labelling information already stored by TRANS is used

! FILE TRANS.SCL (CONTAINING SCALE LIMITS) NOT ON DSK —data limits determined by TRANS on reading in the block output file are used

! FIRST 100 BLOCKS ONLY WILL BE PROCESSED —when specifying blocks by their individual block numbers, a maximum of a hundred blocks may be processed each time the 'GOE' command is used following one or more of the commands 'PRCOLUMN', 'PRPLOT' and 'PLSTRIP'

! FIRST 8 BLKS ONLY WILL BE PROCESSED —a maximum of eight blocks may be processed for 'overlay' plots each time the 'GOE' command is used following the 'PLOVERLAY' command

! INCOMPLETE FILE —for incorrect or incomplete format for parameters and labelling information at beginning of input file (see Appendix E); may occur when creating the input file without running BOMMP (see Section 4)

! LENGTH OF X AXIS MUST BE A +VE INTEGER —s.e.

! LENGTH OF Y AXIS MUST BE A +VE INTEGER LESS THAN 9 —s.e.

! LOWER LIMIT MUST BE LESS THAN UPPER LIMIT —when specifying scale limits

! NOT ENOUGH CORE TO SET UP ARRAYS —when expanding arrays in 'EXPAND'

! ONLY 1 FORM OF O/P TO TTY PER RUN —both tabular and graphical output cannot be obtained on the Teletype by using a single 'GOE' command

! O/P FOR BLK n *NOT STORED IN I/P FILE* —when specifying blocks to be output

! SYNTAX ERROR —typing error or inappropriate response to Teletype message

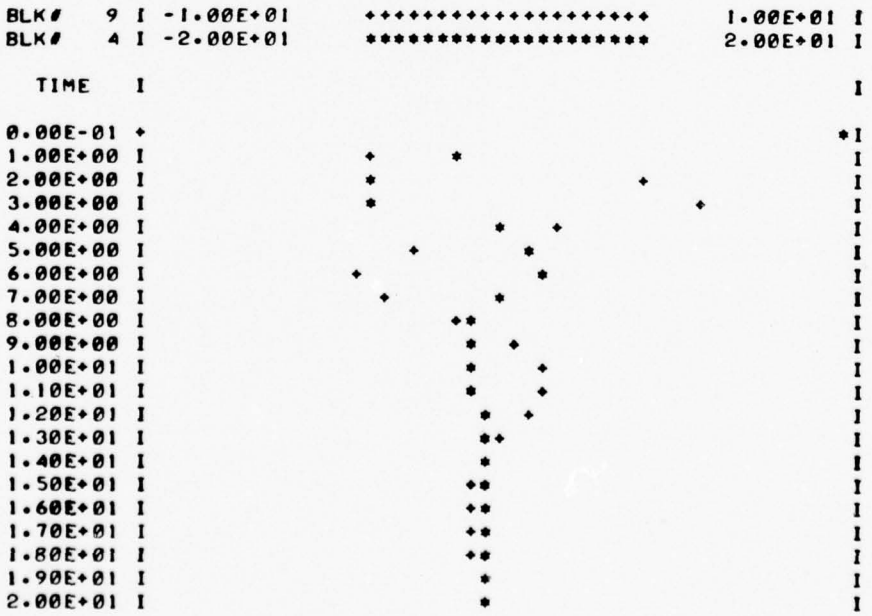
APPENDIX B

Teletype Output

(1) *Tabular Output*

TIME	BLK# 9	BLK# 4	BLK# 48	BLK# 10
0.00E-01	-1.00E+01	2.00E+01	0.00E-01	-1.00E+02
1.00E+00	-3.27E+00	-1.42E+00	9.36E+00	-1.16E+01
2.00E+00	4.50E+00	-6.41E+00	5.52E+00	2.10E+01
3.00E+00	6.06E+00	-6.39E+00	-2.44E+00	3.68E+01
4.00E+00	1.86E+00	1.10E+00	-4.62E+00	3.71E+00
5.00E+00	-2.02E+00	2.01E+00	-2.98E+00	-4.10E+00
6.00E+00	-3.68E+00	2.90E+00	-2.03E-01	-1.41E+01
7.00E+00	-2.69E+00	8.95E-01	1.84E+00	-8.15E+00
8.00E+00	-7.47E-01	-4.38E-01	1.84E+00	-1.49E+00
9.00E+00	8.00E-01	-7.97E-01	1.19E+00	1.60E+00
1.00E+01	1.58E+00	-7.82E-01	3.74E-01	3.16E+00
1.10E+01	1.60E+00	-5.23E-01	-2.92E-01	3.20E+00
1.20E+01	1.10E+00	-1.83E-01	-6.46E-01	2.20E+00
1.30E+01	4.17E-01	1.04E-01	-6.76E-01	8.33E-01
1.40E+01	-1.75E-01	2.63E-01	-4.81E-01	-3.51E-01
1.50E+01	-5.16E-01	2.85E-01	-1.97E-01	-1.03E+00
1.60E+01	-5.80E-01	2.09E-01	5.66E-02	-1.16E+00
1.70E+01	-4.37E-01	9.16E-02	2.08E-01	-8.74E-01
1.80E+01	-2.02E-01	-1.64E-02	2.43E-01	-4.05E-01
1.90E+01	1.96E-02	-8.35E-02	1.89E-01	3.92E-02
2.00E+01	1.62E-01	-1.02E-01	9.25E-02	3.24E-01

(2) *Graphical Output*



APPENDIX C

Line Printer Output

(1) *Tabular Output (File OUTPT.COL printed using 'P' switch)*

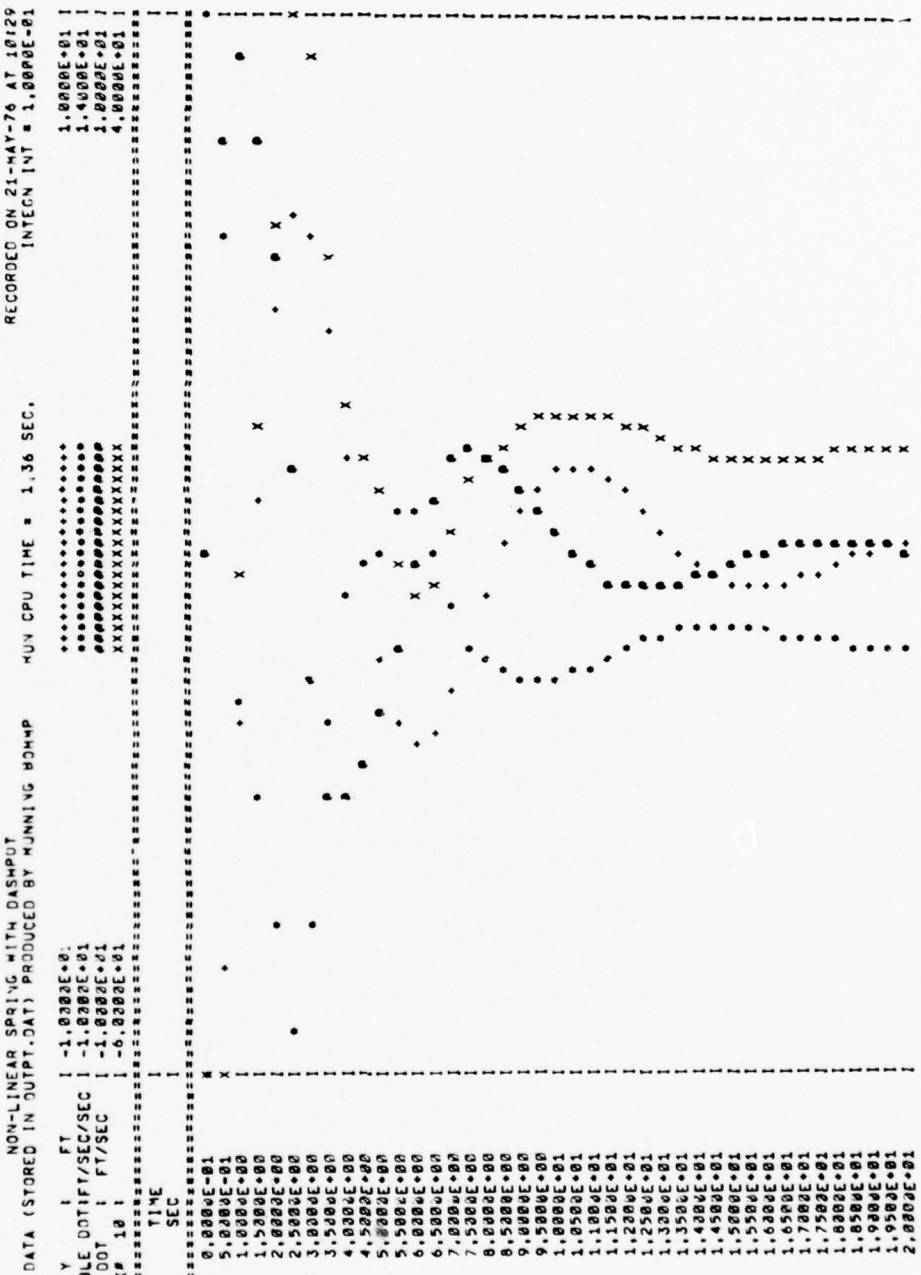
NON-LINEAR SPRING WITH DASHPOT
DATA (STORED IN OUTPT.DAT) PRODUCED BY RUNNING 80HMP

RECORDED ON 21-MAY-76 AT 10:29 INTEGN INT = 1.0000E-01 RUN CPU TIME = 1.56 SEC.

BLK NUMBER = TIME	9 Y FT	4 Y DBLE DOT FT/SEC/SEC	48 Y DOT FT/SEC	10 BLK# 10
0.0000E-01	-1.0000E+01	2.0000E+01	0.0000E-01	-1.0000E+02
5.0000E-01	-7.8020E+00	9.1301E+00	7.7888E+00	-6.1228E+01
1.0000E+00	-3.2697E+00	-1.4199E+00	9.3587E+00	-1.1618E+01
1.5000E+00	1.0834E+00	-3.5982E+00	7.8866E+00	2.1668E+00
2.0000E+00	4.5016E+00	-6.4117E+00	5.5216E+00	2.1316E+01
2.5000E+00	6.3288E+00	-8.7514E+00	1.5770E+00	4.0603E+01
3.0000E+00	6.0589E+00	-6.3873E+00	-2.4440E+00	3.6825E+01
3.5000E+00	4.2110E+00	-1.8077E+00	-4.5356E+00	1.8112E+01
4.0000E+00	1.8573E+00	1.1038E+00	-4.6169E+00	3.7146E+00
4.5000E+00	-2.8736E-01	1.6788E+00	-3.9097E+00	-5.7472E-01
5.0000E+00	-2.2168E+00	2.0119E+00	-2.9791E+00	-4.1308E+00
5.5000E+00	-3.2077E+00	2.9295E+00	-1.7008E+00	-1.1246E+01
6.0000E+00	-3.6830E+00	2.9807E+00	-2.0269E-01	-1.4098E+01
6.5000E+00	-3.4467E+00	2.1054E+00	1.0765E+00	-1.2683E+01
7.0000E+00	-2.6911E+00	8.9477E-01	1.8364E+00	-8.1465E+00
7.5000E+00	-1.7115E+00	-1.0911E-01	1.9843E+00	-3.4232E+00
8.0000E+00	-7.4720E-01	-4.3838E-01	1.8432E+00	-1.4944E+00
8.5000E+00	1.0925E-01	-6.6944E-01	1.5618E+00	2.1851E-01
9.0000E+00	8.0039E-01	-7.9659E-01	1.1911E+00	1.6008E+00
9.5000E+00	1.2942E+00	-8.2981E-01	7.8047E-01	2.5881E+00
1.0000E+01	1.5817E+00	-7.8234E-01	3.7420E-01	3.1633E+00
1.0500E+01	1.6747E+00	-6.7312E-01	8.0110E-03	3.3495E+00
1.1000E+01	1.6003E+00	-5.2314E-01	-2.9241E-01	3.2005E+00
1.1500E+01	1.3953E+00	-3.5334E-01	-5.1199E-01	2.7907E+00
1.2000E+01	1.1021E+00	-1.8258E-01	-6.4564E-01	2.2042E+00
1.2500E+01	7.6292E-01	-2.6392E-02	-6.9694E-01	1.5258E+00
1.3000E+01	4.1668E-01	1.0382E-01	-6.7622E-01	8.3337E-01
1.3500E+01	9.5822E-02	2.0104E-01	-5.9843E-01	1.9164E-01
1.4000E+01	-1.7541E-01	2.6253E-01	-4.8092E-01	-3.5082E-01
1.4500E+01	-3.8162E-01	2.8924E-01	-3.4148E-01	-7.6323E-01
1.5000E+01	-5.1606E-01	2.8507E-01	-1.9663E-01	-1.0321E+00
1.5500E+01	-5.7966E-01	2.5602E-01	-6.0389E-02	-1.1593E+00
1.6000E+01	-5.7956E-01	2.0921E-01	5.6552E-02	-1.1591E+00
1.6500E+01	-5.2735E-01	1.5207E-01	1.4717E-01	-1.0547E+00
1.7000E+01	-4.3716E-01	9.1624E-02	2.0818E-01	-8.7433E-01
1.7500E+01	-3.2402E-01	3.3910E-02	2.3925E-01	-6.4805E-01
1.8000E+01	-2.0227E-01	-1.6378E-02	2.4322E-01	-4.0454E-01
1.8500E+01	-8.4426E-02	-5.6362E-02	2.2458E-01	-1.6885E-01
1.9000E+01	1.9620E-02	-8.3499E-02	1.8912E-01	3.9241E-02
1.9500E+01	1.0301E-01	-9.8440E-02	1.4309E-01	2.0602E-01
2.0000E+01	1.6200E-01	-1.0181E-01	9.2538E-02	3.2399E-01

APPENDIX C—continued

(2) *Graphical Output (File OUTPT.PPL printed using 'P' switch)*



APPENDIX D

Main Program Example for Creating an Input File for TRANS without Running BOMMP

```
C       MAIN PROGRAM

C               EXAMPLE OF A MAIN PROGRAM WHICH USES A PACKAGE OF
C       SUBPROGRAMS THAT WRITE OUTPUT INTO A BINARY FILE SUITABLE
C       FOR PROCESSING BY THE PROGRAM "TRANS"
C               THE SUBPROGRAM PACKAGE CONSISTS OF THE FORTRAN
C       FILE PAKAGE.F4 AND THE MACRO FILE CPU.MAC

C               THE FORTRAN FILE CONTAINS :

C       SUBROUTINE OPINIT(COMENT)
C               - RECORDS PARAMETERS AND LABELLING INFORMATION

C       SUBROUTINE OUTPAK(CLAST,C)
C               - RECORDS OUTPUT FOR EACH INDEPENDENT VARIABLE STEP

C       FUNCTION PAKBIN(VALUE)
C               - USED BY OUTPAK

C       SUBROUTINE FINBIN(TCPU,LOU)
C               - RECORDS CPU TIME AND FINALIZES THE OUTPUT FILE

C               THE MACRO FILE CONTAINS :
C       SUBROUTINE CPU(TCPU)
C                 FIRST CALL PROVIDES CPU TIME SINCE JOB LOGGED IN;
C                 SUBSEQUENT CALLS PROVIDE CPU TIME SINCE PREVIOUS
C                 CALL

C               ARRAYS WITH VARIABLE DIMENSIONS ARE:  CLAST(NBLK),
C       C(NBLK), AND COMENT(2*NBLK), WHERE NBLK IS THE NUMBER OF
C       OUTPUT VARIABLES. AT EACH INDEPENDANT VARIABLE VALUE, PRIOR
C       TO RECORDING THE OUTPUT VALUES, THE INDEPENDANT VARIABLE
C       MUST BE STORED IN "T" AND THE OUTPUT VALUES MUST BE STORED
C       IN THE ARRAY "C". THE ARRAY "CLAST" IS USED INTERNALLY BY
C       THE SUBROUTINE "OUTPAK" TO STORE THE LAST RECORDED OUTPUT
C       VALUES AND THUS SHOULD NOT BE SET BY THE USER
C               FOR THIS EXAMPLE, WHERE NBLK = 7, THE FOLLOWING
C       STATEMENTS MUST BE INCLUDED

      DIMENSION COMENT(14),CLAST(7),C(7),NAM(2)
      COMMON /ZOUTZ/LOU,NBUFF,BUFFER(126),NFLAG,XPC,NBLK
      COMMON /XOUTX/TITLE(12),TZERO,TSAMP,TTOT,DT,T,XLAB(2)

C               THE FOLLOWING VARIABLES ARE SET IN THE DATA
C       STATEMENTS BELOW:
C               XPC     - PARAMETER SUCH THAT A CURRENT OUTPUT
C                         VALUE (Y1) IS RECORDED ONLY IF ABS(Y1-Y0)
C                         IS GREATER THAN (0.01*XPC*Y0), WHERE Y0
C                         IS THE LAST VALUE OF THE OUTPUT VARIABLE
C                         STORED AT A PREVIOUS TIME STEP
C               NBLK    - NUMBER OF OUTPUT VARIABLES
C               LOU     - LOGICAL UNIT NUMBER FOR THE OUTPUT FILE
```

```
C               NAM(2)  - OUTPUT FILENAME (MUST HAVE EXTENSION
C                       NAME "DAT")
C               TITLE(12) - TITLE OF UP TO 60 CHARACTERS
C               COMENT(NBLK) - LABELS OF UP TO 10 CHARACTERS PER
C                       OUTPUT VARIABLE. THE ARRAY IS SET UP SUCH
C                       THAT FOR VARIABLE K, COMENT(2*K-1) AND
C                       COMENT(2*K) CONTAIN THE FIRST AND SECOND
C                       GROUPS OF FIVE CHARACTERS RESPECTIVELY
C               DT      - REFERED TO IN "TRANS" AS THE INTEGRATION
C                       INTERVAL; ITS ONLY PURPOSE IS TO IDENTIFY
C                       THE DATA
C               TZERO   - LOWER INDEPENDANT VARIABLE LIMIT
C               TSAMP   - INDEPENDANT VARIABLE INTERVAL
C               TTOT    - UPPER INDEPENDANT VARIABLE LIMIT
C               NFLAG   - A FLAG SET EQUAL TO 1 INITIALLY
C               XLAB(2) - INDEPENDENT VARIABLE LABEL OF UP TO 10
C                         CHARACTERS
      DATA XPC,NBLK,LOU/0.01,7,1/
      DATA (NAM(J),J=1,2)/'FUNCT.DAT$'/
      DATA (TITLE(J),J=1,9)/'                    VARIOUS FUNCTIONS
     1 OF X'/
      DATA (COMENT(J),J=1,14)/' EXP(2X)  ',' SQUARE  ',
     1' SIN(8X)  ','X*SIN(8X) ','SIN(8X**2)',' SAW   ',
     2' NORMAL  '/
      DATA DT,TZERO,TSAMP,TTOT/0.0,0.0,0.004,2.0/
      DATA NFLAG/1/,(XLAB(J),J=1,2)/'     X     '/

C               OPEN O/P FILE & RECORD THE LABELLING INFORMATION

      OPEN(UNIT=LOU,FILE=NAM)
      CALL OPINIT(COMENT)

C       SUBROUTINE CPU(TCPU) - FIRST CALL
      CALL CPU(TCPU)
C       TCPU CONTAINS CPU TIME SINCE LOGGED IN

C       INITIALIZE INDEPENDENT VARIABLE
      T=TZERO

C       ************************************************************

C               TIME LOOP IN WHICH OUTPUT VALUES ARE CALCULATED
C       AND THEN RECORDED IN THE OUTPUT BINARY FILE WITH THE
C       USE OF THE SUBPROGRAM PACKAGE

10    CONTINUE
      X=T

C               FOR THE INDEPENDENT VARIABLE X, THE FOLLOWING
C       7 FUNCTIONS ARE CALCULATED AND RECORDED

C               (1) EXP(2X)
      C(1)=EXP(2.0*X)
```

```
C               (2) SQUARE WAVE WITH PERIOD 0.4
      PERIOD=0.4
      IF(X.GT.TZERO) GO TO 110
      C(2)=1.0
      SWITCH=TSAMP-PERIOD/2.0+1E-6
110   IF(SWITCH.LT.0) GO TO 140
      IF(C(2).GT.0.0) GO TO 120
      C(2)=1.0
      GO TO 130
120   C(2)=-1.0
130   SWITCH=SWITCH-PERIOD/2.0
      GO TO 110
140   SWITCH=SWITCH+TSAMP

C               (3) SIN(8X)
      C(3)=SIN(8.0*X)

C               (4) X*SIN(8X)
      C(4)=X*C(3)

C               (5) SIN(8X**2)
      C(5)=SIN(8.0*X**2)

C               (6) SAW TOOTH WITH PERIOD 0.4
      IF(X.GT.TZERO) GO TO 150
      C(6)=0.0
      GO TO 170
150   C(6)=X
160   IF(C(6).LT.PERIOD) GO TO 170
      C(6)=C(6)-PERIOD
      GO TO 160
170   CONTINUE

C               (7) STANDARD NORMAL CURVE
      C(7)=0.3989*EXP(-X**2/2.0)

C       OUTPUT VARIABLES AT EACH ITERATION
      CALL OUTPAK(CLAST,C)

      T=T+TSAMP
      IF(T.LE.TTOT+0.5*TSAMP) GO TO 10

C       ************************************************************

C               OBTAIN CPU TIME SINCE LAST CALL (OF CPU), OUTPUT LAST
C       BUFFERFULL, AND CLOSE OUTPUT FILE
      CALL CPU(TCPU)
      CALL FINBIN(TCPU,LOU)
      CLOSE(UNIT=LOU,FILE=NAM)

      END
```

APPENDIX E

Parameters and Labelling Information at Beginning of the Input File

Word Number(s)	Contents
1	Number of output values (N)
2–13	Title of 60 characters
14–17	Date and time of creation of the output file obtained when running BOMMP, which is the input file for TRANS
18	Output time† interval
19	Integration interval used when running BOMMP (for reference only)
20	Upper time† value
21	Lower time† value
22–(24 + 3N)	N + 1 groups of three words, where the first word contains a block number and the second and third words contain the corresponding block label of ten characters specified when running BOMMP (blank if not specified). The first group of three words contains the independent variable block number (−1) and label

† or independent variable.

APPENDIX F

(1) Output Program Structure: Major Subroutines

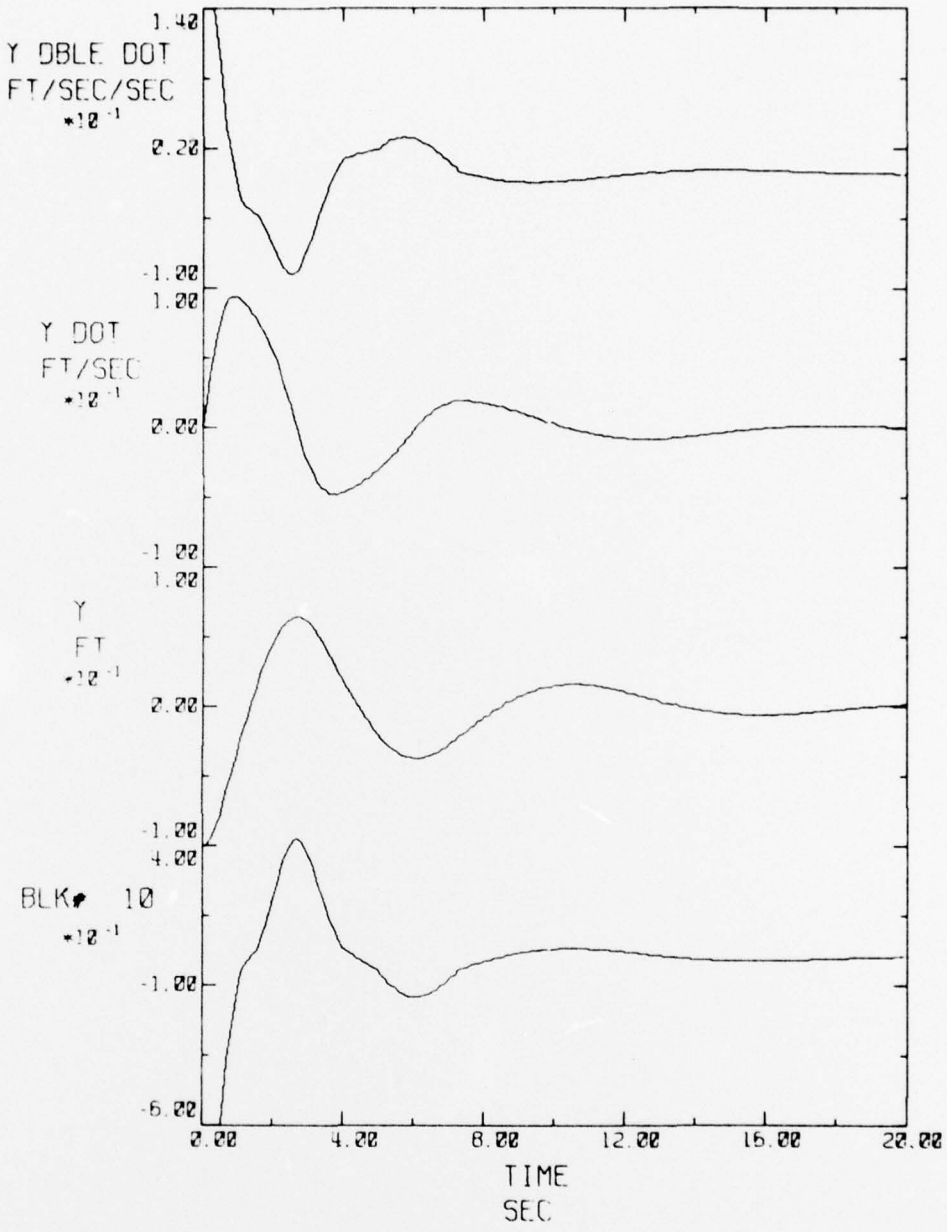
Names	Effect	Section(s) for Reference
TRAN1	Initiates control of the program	2
TRAN2	Controls the flow of the program	2
UNCOM	Unpacks block labels	5.1
INITIA	Unpacks initial set of block values	5.1
UNPACK	Unpacks subsequent sets of block values	5.1
PNTHDS	Produces headings for Teletype and line printer tabular output	2.2, 3.4
PRINT	Produces Teletype and line printer tabular output	2.2, 3.4
HEADNS	Produces headings for Teletype and line printer graphical output	2.2, 3.4
LPLOT	Produces Teletype and line printer graphical output	2.2, 3.4
CAXIS	Calculates origins and scales for incremental plotter output	3.5
STRIP	Produces curves for incremental plotter output	3.5
GRID, PLTHD	Produces axes and labelling information for incremental plotter output	3.5

APPENDIX F—continued

(2) Output Program Structure: Service Subroutines and Functions

Name(s)	Effect	Section(s) for Reference
DING	Sounds bell on the Teletype	—
EXPAND†	Expands an array to the defined number of elements (see Reference 2)	—
FCHECK†	Searches for a file on the disk	—
GTAG, INBLK, ISRT	Accepts lists of block numbers from the Teletype for each form of output	—
ISRCH	Determines position of block in the list	—
MAXMIN, XSCALE, YSCALE, FMIN	Calculates convenient lower and upper limits for graphical output	3.3
MXM, UPDATE	Accepts new scale limits from the Teletype and updates the file TRANS.SCL	3.2
NBR	Used by 'INITIA' and 'UNPACK' to read in a new record of 126 data words	5.1
NUMBER, SYMBOL†	Plots numbers and symbols on the incremental plotter	3.5
OVERLA, OLIST	*Used for 'overlay' plots*	3.5
PLOT*	Plots lines and points on the incremental plotter	3.5
RDC1	Used to determine whether a file may be read from the disk	—
RITCPU	Outputs CPU run time on line printer output	3.4
TD	Enables either Teletype or line printer output to be specified	2.2
TIMOUT	Outputs CPU run time to Teletype	2.1
TXT, APEND, OBTAIN	Accepts labelling modifications from the Teletype and updates the file TRANS.LBL	3.2
XAXIS, YAXIS, COMCHK, NTCK1, NTCK2	Used by 'GRID' to plot axes and labelling information	3.5
ZERO, REMVE	Removes block numbers from lists when output is completed	—

* ARL Library subroutine.
† Written in MACRO-10.

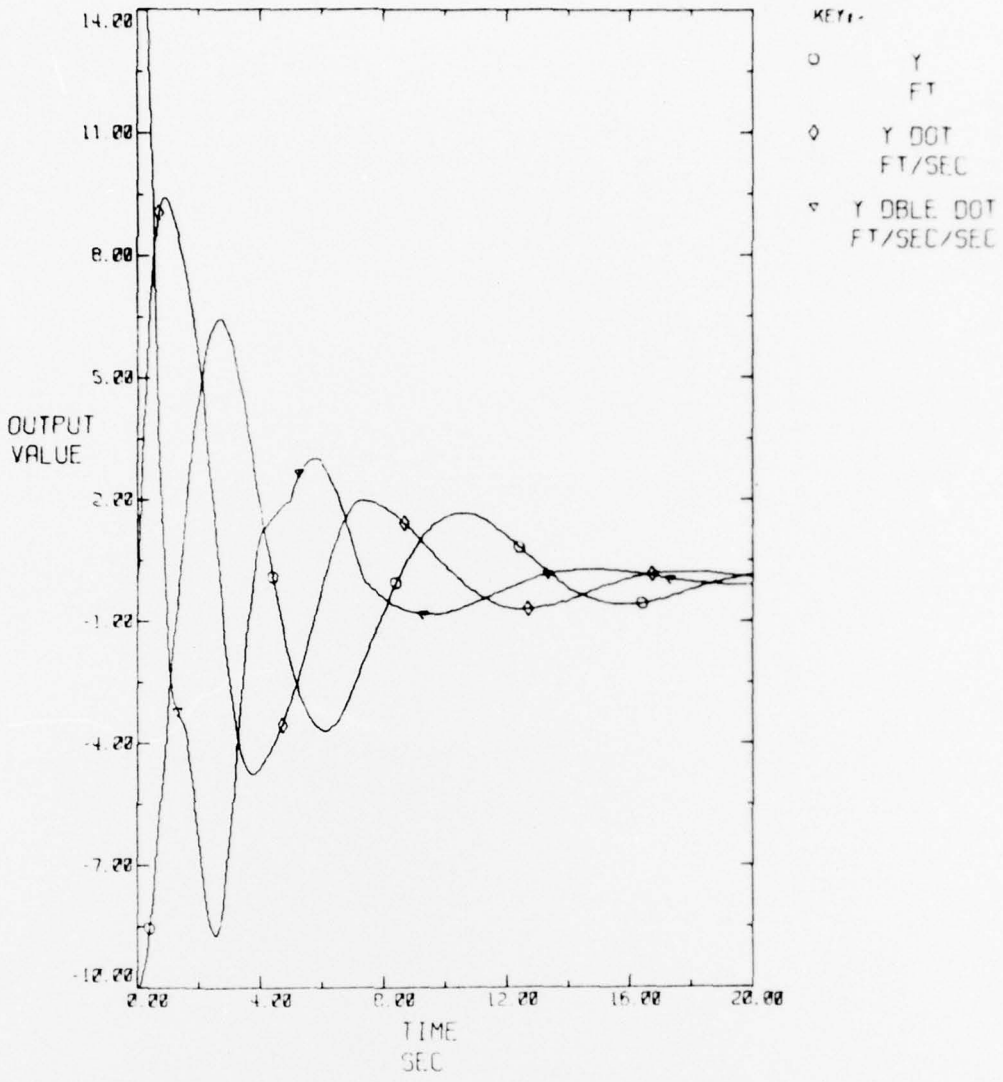


NON-LINEAR SPRING WITH DASHPOT

DATA (STORED IN OUTPT.DAT) PRODUCED BY RUNNING BOMMP

21-May-76 10.23 Δ T = 1.0000E-01 CPU TIME = 1.36 SEC.

FIG. 1a. INCREMENTAL PLOTTER OUTPUT: 'STRIP' PLOTS

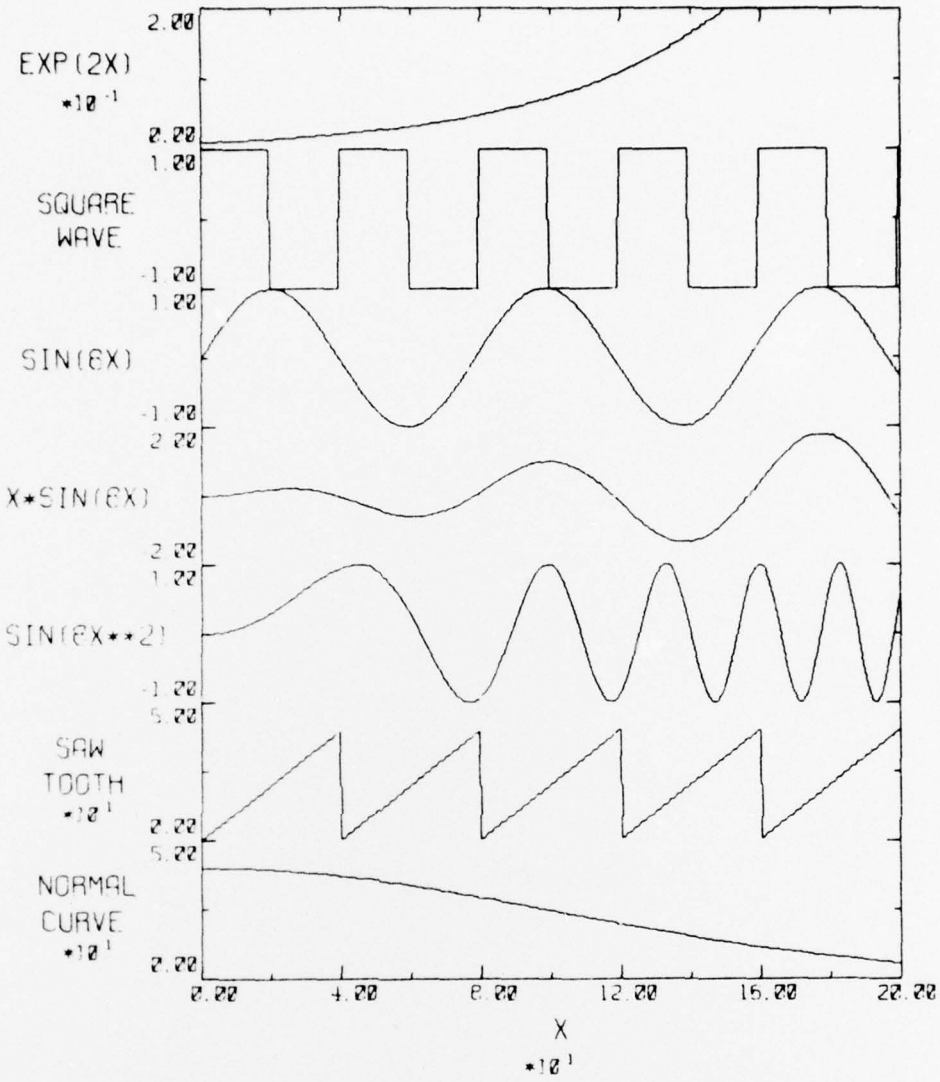


NON-LINEAR SPRING WITH DASHPOT

DATA (STORED IN OUTPT.DAT) PRODUCED BY RUNNING BOMMP

21-May-76 10.29 ΔT = 1.0000E-01 CPU TIME = 1.36 SEC.

FIG. 1b. INCREMENTAL PLOTTER OUTPUT: 'OVERLAY' PLOTS



VARIOUS FUNCTIONS OF X

27-Jul-76 14:02 ΔT = 0.0000E-01 CPU TIME = 1.35 SEC.

FIG. 2. EXAMPLE OF USE OF 'TRANS' WITHOUT RUNNING 'BOMMP' (FILE 'FUNCT. PLT')

DISTRIBUTION

Copy No.

AUSTRALIA

DEPARTMENT OF DEFENCE

Central Office

Chief Defence Scientist	1
Executive Controller, A.D.S.S.	2
Superintendent—Defence Science Administration	3
Superintendent—Assistant to Executive Controller, A.D.S.S.	4
Controller—Service Laboratories and Trials Division	5
Controller—Military Studies and Operational Analysis Division	6
Controller—Programme Planning and Policy Division	7
Central Library	8
S.T.I.B.	9
J.I.D.	10

Aeronautical Research Laboratories

Chief Superintendent		11
Superintendent—Aerodynamics Division		12
Aerodynamics Divisional File		13
N. Gilbert	Co-Authors	14–15
P. Nankivell	Co-Authors	16–17
L. Mitchell	Structures	18
D. A. Secomb	Aerodynamics	19
D. Hatton	Systems	20
R. Whitten	Systems	21
D. Bird	Mechanical Engineering	22
G. Long	Structures	23
B. W. B. Shaw	Aerodynamics	24
D. C. Collis	Aerophysics Group	25–29
Library		30

Central Studies Establishment

Library	31

Materials Research Laboratories

Library, Victoria	32
Library, New South Wales	33
Library, South Australia	34

Weapons Research Establishment

Library	35
T. J. Packer, Aerospace Division	36
P. Goddard, Systems/Analysis Division	37

Universities

Bristol	Library, Engineering Department	139
Cambridge	Library, Engineering Department	140
Manchester	Professor Applied Maths.	141
	Professor N. Johannensen	142
Southampton	Library	143

Colleges and Institutes

Cranfield Institute of Technology	Library	144
Imperial College	Library	145

UNITED STATES OF AMERICA

Australian Defence Research and Development Attache	146
Airforce Flight Dynamics Lab., Wright-Patterson A.F.B., Ohio	147
Arnold Eng. Development Center, Arnold Airforce Station, Tennessee	148
National Technical Information Service, Springfield	149
Library of Congress, Gift and Exchange Department, Washington, D.C.	150
Library, National Bureau of Standards, Washington, D.C.	151
N.A.S.A. Scientific and Information Facility, College Park	152–156
Naval Ordnance Labs., Silver Spring, Md.	157
U.S. Army Ballistics Research Labs., Aberdeen Proving Ground, Md.	158
American Institute of Aeronautics and Astronautics	159
Applied Mechanics Reviews	160
Boeing Co., Head Office, Seattle	161
Lockheed Research Labs. (Palo Alto), Sunnyvale, Calif.	162
Pratt & Whitney Aircraft Division, United Aircraft Corporation, East Hartford	163

Universities

Cornell (New York)	Library, Aero. Labs.	164
Carnegie–Mellon (Pennsylvania)	Library	165
Johns Hopkins	Library	166

Institutes of Technology

California	Library, Guggenheim Aero. Labs.	167

Spares 168–192

(14) ARL/AERO NOTE - 367

DOCUMENT CONTROL DATA

1. Security Grading/Release Limitation

   (a) Document Content: Unclassified

   (b) This Page: Unclassified

2. Document Type/Number
   Aerodynamics Technical Note
   367

3. Document Date
   (11) Dec[redacted]76

(6)

4. Title and Sub-Title: A GENERAL PURPOSE OUTPUT PROGRAM FOR USE IN SIMULATION

(10)

5. Personal Author: P. G. NANKIVELL & N. E. GILBERT

6. Corporate Author(s): AERONAUTICAL RESEARCH LABORATORIES (12) 34 p.

7.

ABSTRACT

The 'block oriented' simulation language CSMP-10(ARL) consists of a modelling program, which performs the model simulation and stores the output in a binary file, and an output program, which prints and plots the character conversion of the binary file. The output program which is described here, is written mainly in FORTRAN IV for a PDP-10 computer and is controlled interactively from a Teletype. Although the output program has been written primarily to be used in conjunction with the modelling program, it may also be used as a general purpose output program by using a supplied subprogram package. The program provides Teletype and line printer output in either tabular or graphical form and incremental plotter output in the form of 'strip' or 'overlay' plots.

8. Computer Program(s)—Titles and Language
   TRANS
   BOMMP (FORTRAN IV)

   008650

9. Descriptors
   Simulation Languages
   Simulation
   Mathematical Models
   Computerized Simulation

11. Cosati Classifications
    0902, 1407, 1201, 1402

12. Task Reference (RD/P)
    RD74/22

10. Library Distribution (Defence Group)
    Central Office Bridges Library
    S.T.I.B.
    J.I.O.
    A.R.L.
    C.S.E.
    M.R.L., Vic., N.S.W., S. Aust.
    W.R.E.
    A.M.T.S.
    A.R.D.U.
    E.D.E.

13. Sponsoring Agency Reference
    A.R.L. File A2/19

14. Cost Code
    51 7740

15. Imprint MELBOURNE—AERONAUTICAL RESEARCH LABORATORIES, 1976